**1 – introdução**

A comercialização de serviços através de meios eletrônicos tem crescido de forma exponencial nos últimos anos. De acordo com a pesquisa realizada pela PWC (PRICEWATERHOUSE COOPERS), uma das maiores prestadoras de serviços profissionais do mundo nas áreas de auditoria e consultoria, o principal fator responsável por esse crescimento é o preço, que tende a ser menor em lojas virtuais. Entretanto, ainda são encontradas certas dificuldades em alguns ramos de serviços, pois, dependendo de qual seja, deverá considerar inúmeras variáveis no momento de sua precificação.

Entre as diversas opções de serviços, a tradução de texto vem crescendo de forma gradativamente ao decorrer dos anos. Segundo pesquisa feita pela Common Sense Advisory(2017), líder de pesquisa e análise independente do setor de tradução, o mercado mundial de serviços de idiomas tem crescido a uma taxa anual de 5,52%, e um fator importante para este crescimento é a globalização mundial. Ao se tratar de assuntos comerciais, as relações lucrativas entre países com linguagens distintas cada vez se torna mais presentes e por isso exigem traduções cada vez mais precisas, que considerem não apenas a transposição de um idioma para outro, mas também a cultura, os aspectos políticos, econômicos e legais do país onde a comunicação será veiculada.

Diversas empresas do ramo de tradução possuem problemas ocultos no momento de calcular orçamentos para seus clientes. O termo “oculto” foi utilizado pois nem sempre está à percepção de todos, por isso até hoje o mercado é maioritariamente baseado na precificação definida por caracteres. Hoje, para a tradução de um arquivo de texto, as variáveis que são consideradas para o cálculo de um orçamento são em sua maioria o número de caracteres, laudas ou palavras que o documento possui, isso resulta na geração de um valor que avalia apenas números e não conteúdo, podendo retornar o mesmo preço para a tradução de um livro infantil e um documento judicial, sendo que claramente a complexidade no processo de tradução de cada um será divergente.

**2 – Objetivo Geral**

Propor e apresentar uma ferramenta seguindo uma nova metodologia para o cálculo de orçamentos de serviços de tradução de documentos de texto, visando o retorno de um valor compatível com o conteúdo presente em cada demanda e não apenas a quantidade de letras, palavras ou laudas. A proposta vem como uma melhoria pois se baseia apenas nas características e *features* que o documento apresenta, com isso o preço da tradução de um artigo científico fica compatível com sua complexidade, assim como uma revista em quadrinhos também ficará pois receberá a mesma análise.

A nova metodologia se torna uma forma mais rigorosa e exata, por isso pode vir a ser aplicada no real mercado do serviço de tradução. Deixando para trás a antiga metodologia que trazia algumas falhas em seu processo.

**2.1 – Objetivo Específico**

**2.1.1 – Pré-processamento**

Elaboração de um algoritmo utilizando a linguagem Python para o pré-processamento dos dados, esta análise lê um diretório de arquivos e retorna a quantidade de:

* Verbos;
* Substantivos;
* Páginas;
* Total de caracteres;
* Média de caracteres por página;
* Palavras;
* Palavras únicas;
* Palavras de um terceiro idioma;
* Caracteres numéricos;
* Índice de Facilidade de Leitura de Flesch.

de cada documento analisado. Após a análise, é criado um *Dataframe* contendo todos estes resultados e uma coluna informando o preço da tradução de cada um destes arquivos, esta coluna será essencial para a aprendizagem de máquina.

**2.1.2 – Processamento**

Foram utilizados os recursos de *Machine Learning* para a análise do *Dataframe* criado anteriormente, o algoritmo considera cada rótulo denominado de *feature* e “aprende” o padrão de precificação de acordo com cada um desses rótulos, ou seja, de qual forma cada um deles influencia diretamente no preço de um arquivo de acordo com a quantidade encontrada nele. Este padrão será aplicado pela Inteligência Artificial (IA) quando um novo arquivo nunca visto por ela for submetido, cada rótulo será analisado e o preço final será calculado de acordo com o que ela aprendeu em seu treinamento.

**3 – Referencial Teórico**

**3.1 – Python**

Para o desenvolvimento da aplicação, foi utilizada a linguagem de programação Python 3.7.0. A escolha desta linguagem foi feita devido ao fato de ser a que melhor atende à proposta do projeto por ser:

* Uma linguagem de alto nível muito acessível. Foi criada com a intenção de ser facilmente entendida por crianças (VAN RUSSON, 1999).
* Portadora de uma sintaxe excepcionalmente legível e com transparência sintática e semântica.
* Uma linguagem interpretada, o Python é cabível para exploração interativa. O programador pode experimentar e receber feedbacks imediatos.
* Portadora de um extenso acervo de bibliotecas gratuitas que oferecem inúmeras funções e métodos aplicáveis de acordo com a necessidade de cada projeto. Como por exemplo a biblioteca NLTK, que trabalha com Processamento de Linguagem Natural (PLN) e tem como objetivo a compreensão automática de linguagens humanas, de maneira que podem ser manipuladas por computadores.

De acordo com Mark Lutz (*Programming Python*, 2001), existem quatro motivos que resumem as razões para a linguagem Python ser utilizada, são eles:

1. **Qualidade**, pois ele torna fácil a maneira de codificar *softwares* que poderão ser reutilizados e mantidos;
2. **Produtividade**, pois a linguagem é otimizada para desenvolvimento ágil;
3. **Portabilidade**, pois a maioria dos programas em Python conseguem executar sem alterações em quase todos os computadores utilizados hoje em dia;
4. **Integração**, pois a linguagem foi projetada para ser integrada com outra ferramentas.

**3.2 – Natural Language Toolkit (NLTK)**

O NLTK é um conjunto de bibliotecas criada por Steven Bird, Edward Loper e Ewan Klein no ano de 2001 e tem como funcionalidade processamento textual, sua criação teve como base 4 (quatro) objetivos primários, são eles:

1. **Simplicidade**: Fornecer um framework intuitivo junto a blocos de construção substanciais, dotando os usuários de um conhecimento prático de PLN sem prender-se nas tediosas tarefas de "arrumação da casa" geralmente associadas com o processamento de dados linguísticos anotados.
2. **Consistência**: Fornecer um framework unificado com interfaces e estruturas de dados consistentes, e nomes de método facilmente conjecturáveis.
3. **Extensibilidade**: Fornecer uma estrutura que proporcione que novos módulos possam ser incluídos facilmente, incluindo implementações alternativas a diversas abordagens para uma mesma tarefa.
4. **Modularidade**: Fornecer componentes que possam ser utilizados sem a necessidade de compreender o restante do toolkit.

**3.2.1 – Tokenização**

Para melhor análise de informações contidas em documentos, existe a função *Tokenize* da biblioteca NLTK para efetuar a quebra de textos em palavras. Esta função cria um *array* com todas as palavras separadamente, sendo possível analisar e classificar uma por uma de acordo com sua categorização.

from nltk.tokenize import word\_tokenize   
tokenized\_word=word\_tokenize(new\_string)

Algoritmo 1 - Função Tokenize da biblioteca NLTK.

**3.4 – Categorização e classificação de palavras**

Para a classificação das palavras separadas no processo de Tokenização, existe a função *Pos\_tag* da biblioteca NLTK.

classificado=nltk.pos\_tag(tokenized\_word)

Algoritmo 2 - Função *Pos\_tag*

Esta função executa a chamada “*parts of speach tagging*”, ela também é conhecida por conter categorias lexicais pelo fato de analisar cada palavra e categorizar através de suas tags específicas que determinam e classificam cada palavra existente em um texto, são elas as classificações:

* CC: Conjunção Coordenativa
* CD: Dígito Candinal
* DT: Determinador
* EX: Existencial (ex: ‘existe’, ‘há’)
* FW: Palavra Estrangeira
* IN: Preposição Conjunta/Conjunção Subordinada
* JJ: Adjetivo (ex: ‘grande’)
* JJR: Adjetivo Comparativo (ex: ‘maior’)
* JJS: Adjetivo Superlativo (ex: ‘o maior’)
* LS: Marcador de Lista (ex: ‘1)’ )
* MD: Modais (ex: ‘dever’, ‘poder’)
* NN: Substantivo Singular (ex: ‘mesa’)
* NNS: Substantivo Plural (ex: ‘mesas’)
* NNP: Substantivo Próprio Singular (ex: ‘Paulo’)
* NNPS: Substantivo Próprio Plural (ex: ‘Americanos’)
* PDT: Predeterminador (ex: ‘todas as crianças’)
* POS: Finais Possessivos
* PRP: Pronome Pessoal (ex: ‘eu’, ‘ele’, ‘ela’)
* PRP$: Pronome Possessivo (ex: ‘meu’, ‘dela’, ‘dele’
* RB: Advérbio (ex ‘silenciosamente’)
* RBR: Advérbio Comparativo (ex: ‘melhor’)
* RBS: Advérbio Superlativo (ex: ‘educadíssimo’)
* RP: Prático
* TO: Ações (‘ir à loja’)
* UH: Interjeições (ex: errrrrrrrm)
* VB: Verbo
* VBD: Verbo Pretérito
* VBG: Verbo, Gerúndio/Particípio Presente
* VBN: Verbo, Particípio Passado
* VBP Verbo, 1ª pessoa
* VBZ: Verbo, 3ª pessoa
* WDT: Determinador-WH (ex: ‘qual’)
* WP: Pronome-WH (ex: ‘quem’, ‘o que’)
* WP$: Pronome-WH Possessivo (ex: ‘de quem’)
* WRB: Advérbio-WH (ex: ‘onde’, ‘como’)

**3.5 – Rede Neural artificial (RNA)**

NELSON & ILLIGNWORTH (1990), definem as RNA’s como sendo técnicas inovadoras de processamento de informações, que têm como objetivo básico simular, em computadores, o funcionamento do sistema nervoso biológico.

Já Murillo Grübler (2018) apresenta uma definição mais técnica para as RNA’s como uma estrutura complexa interligada por elementos de processamento simples (neurônios), que possuem a capacidade de realizar operações como cálculos em paralelo, para processamento de dados e representação de conhecimento.

A Rede Neural Artificial é uma abstração da Rede Natural Biológica, ela serve de modelo para o aprendizado e a resolução de problemas complexos, tais como reconhecimento de fala e imagens, processamento de linguagem natural e previsão de valores.

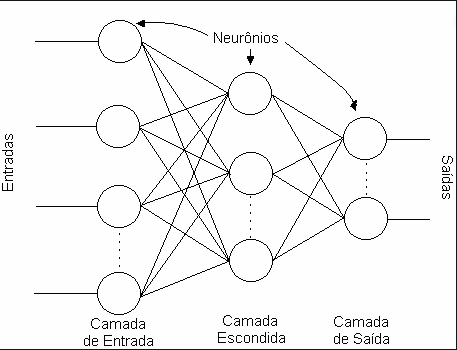


Figura 1 - Arquitetura *Perceptron*. Fonte – Fabio Mendes Soares

**3.5.1 – *Multi-Layer Perceptron* (MLP)**

O modelo de *Multi-Layer Perceptron* surgiu como uma evolução do modelo *Perceptron* criado em 1958 por Frank Rosenblatt (Van Der Malsburg, 1986), que é um dos modelos mais antigos modelos e lida com um único neurônio, classificando o resultado de forma linear. Já o MLP apresenta um modelo de *Perceptron* em múltiplas camadas. O grande diferencial que o MLP apresentada em relação ao *Perceptron* simples é a sua capacidade de solucionar problemas não linearmente separáveis, ou seja, que são necessárias mais de uma reta para que os grupos possam ser separados.

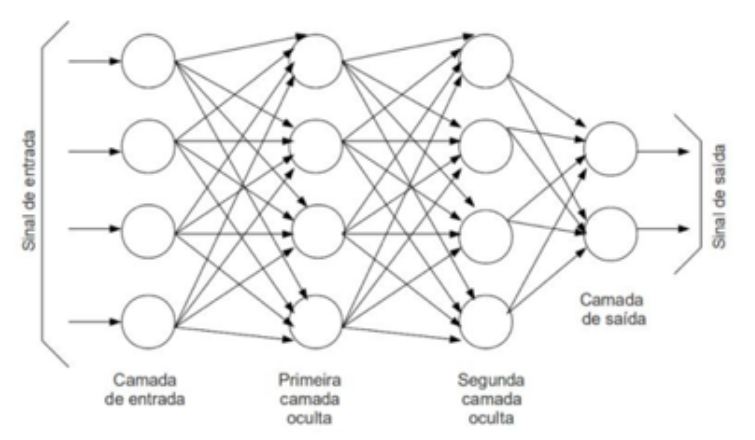
Este modelo possui uma arquitetura em camadas, que podem conter números variados de neurônios em cada uma. Elas são organizadas de forma que os neurônios de cada camada recebe como entrada a saída da camada anterior, o que é chamado de *feed-forward.*

Figura 2 - Arquitetura MLP (*Multi-layer Perceptron*). Fonte - Victor Gutemberg Oliveira Marques

**3.5.2 – Simple Linear regression**

A *Simple Linear Regression* é um método estatístico para obter uma fórmula com objetivo de prever valores de uma variável para outra onde há uma relação casual entre essas duas variáveis.

A fórmula para um gráfico linear se entende como  
 y = a + bx  
onde y é uma variável dependente e x é uma variável independente. Este modelo encontra a função linear de um problema e, com maior precisão possível, prevê o valor da variável dependente como função da variável independente.

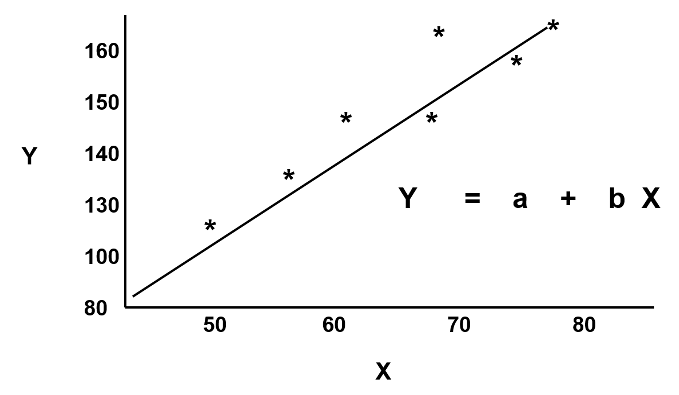


Figura 3 – *Simple Linear Regression*. Fonte - Boston University School of Public Health

**3.6 - Índice Flesch**

Para ser possível fazer a precificação de todos os arquivos que foram utilizados para o treinamento da rede neural MLP de forma ágil, foi necessário levar como principal preocupação a complexidade destes arquivos, pois é o principal fator focado pelo algoritmo da IA. Sendo este o foco, foi selecionado para ser utilizado o Índice de Facilidade de Leitura de Flesch.

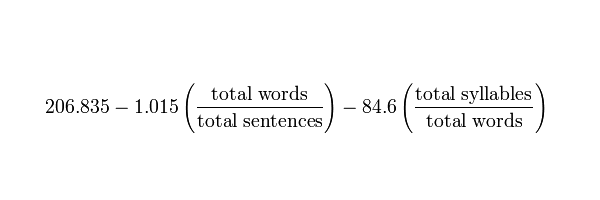
O Índice Flesch é uma das fórmulas de legibilidade mais utilizada atualmente, sendo considerada adequada para todos os tipos de texto, ela estima o grau de leiturabilidade de um texto levando em consideração o tamanho médio das frases em relação ao tamanho médio das palavras, que são as principais variáveis que podem afetar a facilidade de leitura de um texto.O índice de facilidade de leitura de Flesch classifica o texto em escala de 0 a 100 pontos e o seu resultado classifica o texto de acordo com a dificuldade de leitura (Tabela 1), quanto menor for o resultado do cálculo, mais difícil é o nível de legibilidade do texto.

Tabela 1 - Escore Flesch distribuído de acordo com nível de escolaridade

|  |  |
| --- | --- |
| **Escala Flesch** | **Nível de legibilidade** |
| 0-29 | Muito Difícil |
| 30-49 | Difícil |
| 50-59 | Razoavelmente Difícil |
| 60-69 | Padrão |
| 70-79 | Razoavelmente Fácil |
| 80-89 | Fácil |
| 90-100 | Muito Fácil |

Fonte - LUCAS LOBATO

Os valores que se encaixam na Tabela 1 são encontrados através da seguinte fórmula:



206,836 menos 1,015 vezes a divisão do total de palavras pelo total de frases menos 86,6 vezes a diferença entre o total de sílabas e o total de palavras.

**4 – Trabalhos Relacionados**

No estudo feito por Cássio Oliveira Camilo e João Carlos da Silva, estudantes do Instituto de Informática Universidade Federal de Goiás, foi apresentado de forma aprofundada os conceitos fundamentais da Mineração de Dados. A mineração de dados é uma das tecnologias mais promissoras da atualidade pelo fato de dezenas, e às vezes centenas de milhões de reais serem gastos pelas companhias na coleta dos dados e por fim nenhuma informação útil é identificada.

No estudo é trazido as principais áreas nas quais a Mineração de Dados é aplicada de forma satisfatória como por exemplo na Retenção de Clientes, Bancos, Cobranças, Telemarketing, Eleitoral, Medicina, Segurança, RH, Agência de Viagens e muito mais.

A mineração de Dados tornou-se uma ferramenta de apoio com papel fundamental na gestão da informação dentro das organizações. A maneira tradicional de manipulação de dados e análise de informações tornou-se inviável devido ao grande volume de dados, por isso, descobrir padrões implícitos e relacionamentos em repositórios que contêm um grande volume de dados de forma manual, deixou de ser uma opção.

Não há dúvidas de que é uma área extremamente promissora e que, apesar dos resultados já obtidos, ainda tem muito a oferecer.

**5 – Metodologia**

**5.1 – Desenvolvimento da aplicação**

Conforme citado no tópico 4.1, a aplicação será desenvolvida utilizando a linguagem Python na versão 3.7.0 e aplicando essencialmente a biblioteca NLTK para o processamento de texto, classificação, tokenização, análise e raciocínio semântico.

A IDE (Integrated Development Environment) selecionada para o desenvolvimento do projeto foi a Visual Studio Code na versão 16.3.4. Sua seleção foi devido ao fato de ser um editor de código-fonte desenvolvido pela Microsoft que inclui suporte para depuração, controle Git incorporado, realce de sintaxe, complementação inteligente de código, *snippets* e refatoração de código. É um software livre e de código aberto.

**5.2 – Precificação dos Arquivos**

Para o treinamento da rede neural, o dataframe gerado pelo algoritmo com as *features* necessita conter o preço de cada arquivo, desta forma a própria rede aprende o peso de cada *feature* e no momento da precificação de um novo arquivo ele aplica este peso e encontra valores baseados na mesma verificação feita para os outros arquivos. Como o foco da IA é a complexidade, foi selecionado o Índice Flesch para encontrar o nível de legibilidade de cada arquivo, este índice foi utilizado como um fator importante no momento de precificar os arquivos para o treinamento do algoritmo.

Foram determinados pesos para cada feature, com isso os valores que o algoritmo encontra para cada arquivo, serão multiplicados por seus respectivos pesos, e por fim o resultado da soma das multiplicações será multiplicado pelo fator de acordo com o Índice.

**5.3 – Funcionamento da aplicação**

**5 – Resultados Encontrados**

**5.1 – Tratamento de Dados**

Para o treinamento da IA, é necessário um Dataframe com as informações de cada documento analisado pelo código. Para gerar este dataframe, foi utilizada a biblioteca Pandas que possui funções destinadas à criação e manipulação de dataframes.

Algoritmo 3- Import da biblioteca pandas.

import pandas as pd

O arquivo gerado apresenta todas as *features* que serão consideradas pela IA como colunas e cada linha representa um arquivo.

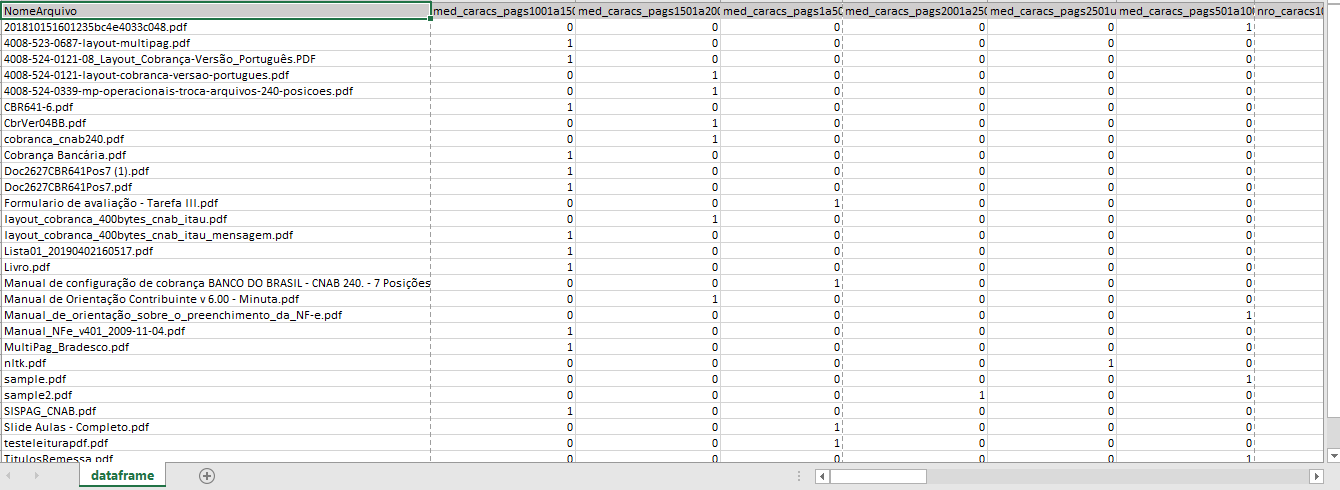


Figura 2 - Imagem de parte do Dataset gerado pelo algoritmo

Para ser gerado o dataframe, foi montado primeiro um dicionário com todas as *features* que seriam analisadas em cada arquivo já com o 0 preenchido em todas. Para melhor entendimento, foi definido 0 para “Não” e 1 para “Sim”.

Algoritmo 4 - Preenchimento do Dataset com o número 0.

data\_dict={'NomeArquivo':'0', 'verbos':'0', ...}

A análise dos arquivos foi feita dentro da estrutura de repetição FOR que percorre uma lista com todos os arquivos do diretório em foi determinado.

Algoritmo 5 - Repetição para analisar o arquivo PDF.

path = "./diretorio/"  
dirs = os.listdir(path)  
df = pd.DataFrame()  
for file in dirs:  
 ...

Como é possível verificar no trecho do código acima, foi criado um Dataframe vazio, que será acrescido de informações no decorrer das análises dos arquivos.

As informações do dicionário são alteradas de 0 para 1 à medida em que o documento é analisado e se encaixa nas condições determinadas para cada análise. No exemplo abaixo, caso o arquivo tenha um número de páginas entre 0 e 10, o dicionário é alterado para 1 na posição desta determinada *feature*.

Algoritmo 6 - Preenchendo o DataSet com o número 1 caso a condição for atendida

if pdf.getNumPages() >= 0 and pdf.getNumPages() <= 10: data\_dict.update({'nro\_pags':'1'})

Após passar em todas as condições e chegar ao fim das análises, todo o dicionário foi atualizado com as *features* pertencentes a cada documento.   
Ao fim, o dicionário inteiro é inserido no Dataframe através do método append() da biblioteca Pandas.  
Com o Dataframe populado, é gerado o documento em formato .csv contendo as linhas e colunas que serão analisadas pelo algoritmo da Inteligência Artificial na segunda parte do código.

Algoritmo 7 - Gerando arquivo Dataframe com a extensão CSV

*#POPULANDO O DATAFRAME COM AS INFORMAÇÕES DO DICIONÁRIO*  
df = df.append(data\_dict, *ignore\_index*=True)

*#CRIAÇÃO DO DATAFRAME COM OS DADOS DO DICIONÁRIO*df.to\_csv('dataframe.csv', *sep*=';',*encoding*='cp1252')

**5.2 – Treinamento da Rede Neural**

**6 – Conclusão**

O desenvolvimento do presente estudo possibilitou uma análise comparativa entre o modelo atual de precificação de serviços de tradução com o novo modelo proposto que se baseia na complexidade da leiturabilidade do arquivo.

O estudo mostrou como o modelo atual utilizado pela maioria das prestadoras de serviços de tradução acaba sendo falha por levar em consideração apenas características quantitativas como quantidade de caracteres e laudas, vide os resultados encontrados, que mostram que arquivos com maior quantidade de caracteres e laudas não são necessariamente serviços mais complexos comparados a arquivos menores, com o modelo proposto propomos um preço mais condizente com a complexidade do serviço.

**7 – Referências Bibliográficas**

RODRIGUEZ, M.; PETERSON, R. M.; KRISHNAN, V. Social Media's Influence on Business-To-Business Sales Performance. Journal of Personal Selling & Sales Management, v. 32, n. 2, p. 365-378, Summer 2012 2012. ISSN 0885–3134 (print) / ISSN 1557–7813 (online). Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=78110794&site=ehostlive >. Acesso em: 09 abr. 2019.

PWC (PRICEWATERHOUSECOOPERS) (2016), “Total Retail Brasil 2016”. Disponível em https://www.pwc.com.br/pt/setores-de- atividade/varejo-e-consumo/assets/2016/total\_retail\_16\_brasil.pdf>. Acesso em 09 abr. 2019.

DINO DIVULGADOR DE NOTÍCIAS (2017), “Com tecnologias cada vez mais avançadas, o mercado de serviços de tradução não para de crescer”. Disponível em <https://exame.abril.com.br/negocios/dino/com-tecnologias-cada-vez-mais-avancadas-o-mercado-de-servicos-de-traducao-nao-para-de-crescer-shtml/>. Acesso em 15 out. 2019.

Van Rossum, 1999 van Rossum, G. (1999). Computer programming for everybody. Technical report, Corporation for National Research Initiatives. <http://www.python.org/doc/essays/ cp4e.html> Acesso em 19 out. 2019.

CÁSSIO OLIVEIRA, C.; CARLOS DA SILVA, J.; Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas. Agosto, 2019. Disponível em: <https://rozero.webcindario.com/disciplinas/fbmg/dm/RT-INF\_001-09.pdf> Acesso em 20 out. 2019

VICTOR GUTEMBERG OLIVEIRA MARQUES.(2017) “AVALIAÇÃO DO DESEMPENHO DAS REDES NEURAIS CONVOLUCIONAIS NA DETECÇÃO DE OVOS DE ESQUISTOSSOMOSE”. Disponível em <https://www.cin.ufpe.br/~tg/2017-1/vgom\_tg.pdf>. Acesso em 15 nov. 2019.

LUCAS LOBATO, BEATRIZ SANTANA CAÇADOR , MARIA FLÁVIA GAZZINELLI .(2013) “LEGIBILIDADE DOS TERMOS DE CONSENTIMENTO LIVRE E ESCLARECIDO EM ENSAIOS CLÍNICOS”. Disponível em <http://www.scielo.br/pdf/bioet/v21n3/a20v21n3.pdf>. Acesso em 15 nov. 2019.

Lutz, Mark. *Programming python*. " O'Reilly Media, Inc.", 2001.

NELSON, Marilyn M. & ILLINGWORTH, W. T. A Praticai Guide to Neural Nets. Addison-Wesley Publishing Company, Inc. USA, 1991.

Grübler, Murillo. “Entendendo o funcionamento de uma Rede Neural Artificial”. Disponível em: <https://medium.com/brasil-ai/entendendo-o-funcionamento-de-uma-rede-neural-artificial-4463fcf44dd0>

Van Der Malsburg, C. "Frank Rosenblatt: principles of neurodynamics: perceptrons and the theory of brain mechanisms." *Brain theory*. Springer, Berlin, Heidelberg, 1986. 245-248.

Simple linear regression. Disponível em: < http://www.statstutor.ac.uk/resources/uploaded/slregression.pdf>